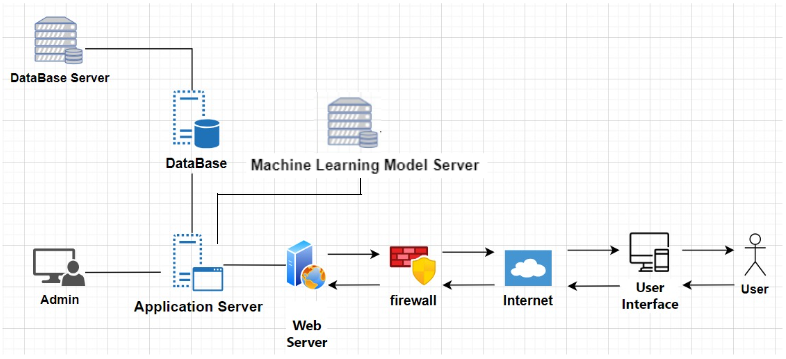
Detailed Design

Physical Architecture



User Interface - Users will access our website through their web browsers on their devices, such as desktops, laptops, tablets, and smartphones.

Web Server - The frontend code for our website, such as HTML, CSS, and Javascript, will be hosted on a web server.

Internet Connection - Our servers will remain connected to the internet so that they can receive requests from users and access our databases.

Firewall - The firewall will protect our servers and data from unauthorized access and attacks.

Application Server - The backend code for our website will be hosted on an application server. This server will handle requests from users and the frontend, process data, perform calculations, interact with the database and machine learning server, and generate dynamic content.

Machine Learning Model Server - A dedicated server responsible for hosting and serving the machine learning models. This server will handle the prediction requests from the application server and return the predicted probabilities.

Database & Database Server - Our website data will be organized into a database which will be stored and managed on a database server. Our database will consist of tables storing student data, school information, degree details, and any other relevant information. We will interact with the database and perform operations such as inserting, updating, deleting, and querying the data.

Admin - Our admin will manage the content displayed on our website, including adding, updating, or removing information about degrees, schools, and other relevant data. This also includes performing administrative tasks related to the database, such as backups, restoring, and structural database modifications.

Component Diagram



Frontend

1. Submission Page - This page allows past students to submit their academic details.
2. Search Tool Page - Users can utilize this page to search for acceptance requirements based on various criteria such as degree type, institution, or required grades.
3. Results Page - Upon submitting a search query, users are directed to this page, where they can view a list of degree programs matching their criteria.
4. Detailed Results Page - This page provides additional details about admission requirements for the schools or degrees selected from the search results page.

Backend

1. Submission Handling - The backend will be responsible for processing and storing user submissions.
2. Search Query Processing - Backend logic will execute search queries received from the frontend, retrieve relevant data from the database, and return search results to the user interface.
3. Data Retrieval for Further Information - When users request additional information about the admission requirements for specific degrees or schools, the backend will retrieve the relevant data from the database and Machine Learning Server and format it for its presentation on the frontend.

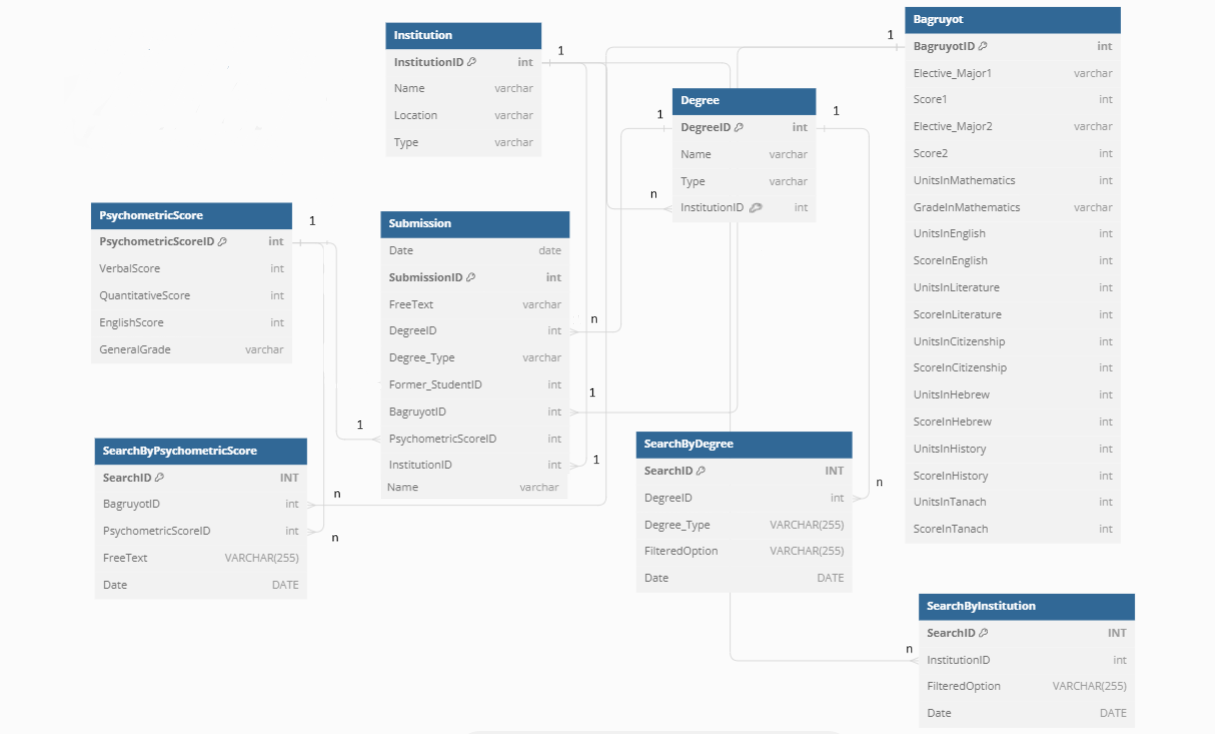
Database

1. Submission Table - This table stores user submissions.
2. Institution Table - Contains basic information about institutions.
3. Degree Table - Stores data about degree programs offered by institutions.
4. Search History Tables - These tables record user search activity.

Machine Learning Server

1. Model Hosting & Execution - The ML server will host and execute machine learning models responsible for predicting admission probabilities and extracting important features in the admission process.
2. Integration with the Backend - It will receive data requests from the backend for prediction or feature extraction tasks and return the results back to the backend.

ERD



ERD Explanation

In our database, we'll have several key tables to manage our data. Our student submissions table will house vital information related to each student's submission, including their bagruyot scores, psychometric scores, and additional details contributing to their admission to a school and degree program. To streamline our database structure, we've opted to store bagruyot scores and psychometric scores in separate, more detailed tables.

Additionally, we'll maintain separate tables for institution data and information about the degrees offered by each institution. These tables will contain basic information, and ensure ease of access when retrieving relevant data.

We will track user searches in three distinct tables. This approach involves recording the search date, specific information sought, and any applied filters. By doing so, we can effectively monitor searched data and trends over time, providing valuable insights into consistently sought-after topics. This information will guide us in optimizing our website to better meet user needs and preferences.

UML Diagrams of Use Cases

1.



2.



3.



Algorithms

Submission Tracking Algorithm

This algorithm uses the collected submissions data from past students. It will calculate the frequency of submissions over time and evaluate the influxes of user engagement. These insights will help us enhance our user engagement and overall optimize our business.

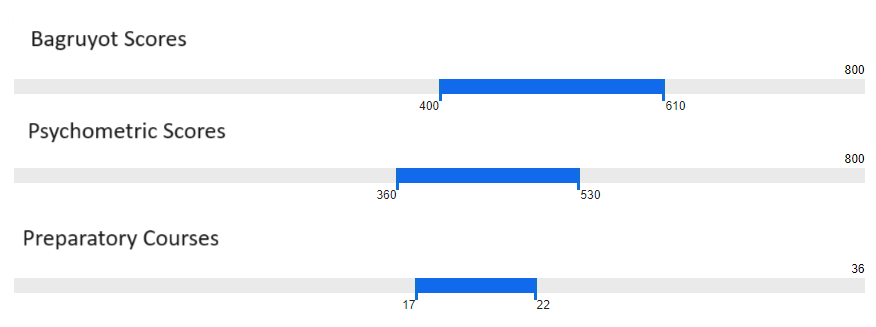
Search Behavior Analysis Algorithm

This algorithm analyzes past searches from the database to better understand our user behavior. It analyzes search dates, filter usage, institutions and degrees searched. We will calculate the frequency of each item to determine the popular choices. This analysis will provide insights into how we can enhance the search functionality and user experience continually.

When a user makes a search based on institution, degree, or grades, we will present their results in various formats on the detailed results pages, so that the information and insights are digestible and clear for users. To do this we will implement the following algorithms:

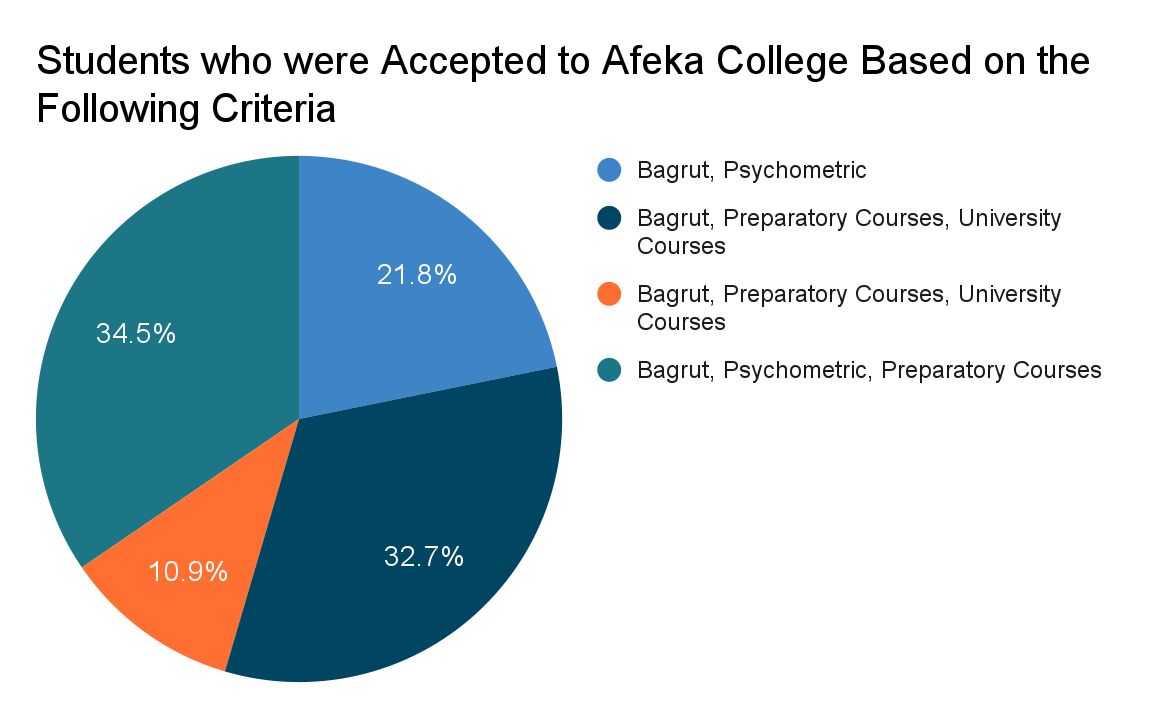
Calculating Averages, Ranges, and Distribution Algorithms

This algorithm computes the average, range, and distribution of different scores for each academic criteria for each institution or degree. To calculate the averages, it aggregates scores of particular criterias by summing them up and dividing them by the total number of submissions. Score ranges are determined by finding the minimum and maximum values within each criterion. For the distribution we get the frequency of scores falling within specific intervals or bins for each criteria.



Aggregating Acceptance Requirements Algorithm

This algorithm is designed to showcase the academic requirements that lead to the acceptance of past students by institutions and specific degrees. It starts by identifying all possible combinations of acceptance criteria. Next, it tallies the occurrences of each combination among the accepted submissions and computes the average scores for each combination. The results are then presented to the user via a pie chart, illustrating the distribution of the various combinations of acceptance requirements.



Academic Feature Importance Algorithm

This algorithm utilizes machine learning techniques to determine the most influential academic features in the admissions process of a school or degree program. It starts by using the submission data of past applicants. Then, it preprocesses the data, which includes feature engineering and model training. The algorithm then analyzes feature importances to identify which factors contributed most significantly to the admissions decisions. These insights will help prospective students understand which requirements are most important for acceptance.

Predicting Acceptance Algorithm

This algorithm utilizes machine learning techniques to predict the probability of acceptance into a program based on a potential student’s grades. It starts by using the submission data of past applicants. Then, it preprocesses the data, which includes cleaning and transformation, the machine learning models are then trained to learn the patterns from the grades data. Once trained, the models will predict the probability of acceptance for new applicants. The algorithm will aid potential students in making informed decisions about their application success rates.